Advancing Interactivity in Computer Games through Ontology and Mixed-Initiative

Liam Doherty, Max Whitney, Vive Kumar

Simon Fraser University, Canada
ldoherty@sfu.ca, mwhitney@sfu.ca, vive@sfu.ca

Abstract

Providing an immersive and interactive game environment requires rich and detailed interaction, both between the player and the game world, and between game world agents. The concepts of mixed-initiative and ontology are applied to this challenge, and their usefulness to computer game design is discussed. An ontology-based system for enabling increased interactivity through mixed initiative is proposed, and the possible effects of mixed-initiative on game play and agent AI design are considered.

A Definition of Mixed-Initiative in Computer Games

A general definition of mixed-initiative is still a work-in-progress for the field. Mixed-initiative theory has been applied to the fields of educational applications, planning systems, machine learning, and others, and in each case mixed-initiative means something somewhat different. Mixed-initiative is usually applied to problem solving, where it can mean (according to James F. Allen) "…a flexible interaction strategy, where each agent can contribute to the task what it does best. …[I]n the most general cases, the agents' roles are not determined in advance, but opportunistically negotiated between them as the problem is being solved." (Hearst, 1999) In applying mixed-initiative theory to games, we assume that both the task (or problem) and roles of the agents involved in a given interaction can be negotiated.

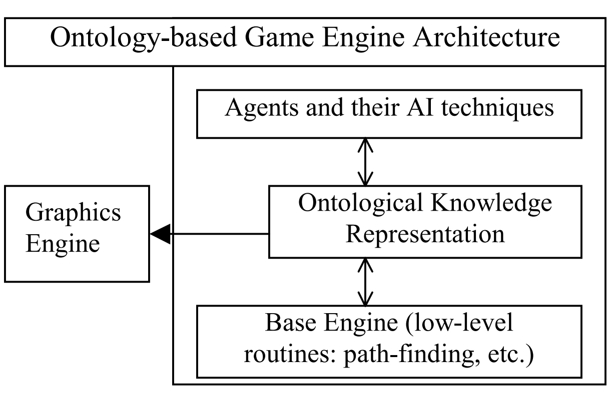
For the purposes of this paper, an interaction between agents (players, non-player characters, or game objects – anything that can interact with anything else in the game world) will be considered a mixed-initiative interaction if the following conditions are met. Firstly, all participating agents must have some set of goals that motivate them to take initiative to enter into an interaction. Once the interaction begins, it should include negotiation over the collective (group) goal. Secondly, all agents must have the ability to initiate interactions with some other agents. Finally, the agent must be able to take some actions that are not explicitly defined by another agent (for example, if the player tells another agent to cut down a tree, the agent will choose which tree to cut down and handle the path finding to get there.) In other words, the agent must have some of its own knowledge about the world that informs its interpretation of instructions or commands it receives. It is important to keep in mind that mixed-initiative can happen between the player and computer characters, between computer characters, or both (since much more than two agents may be part of a mixed-initiative interaction.) Of course, not all interactions in a game will be mixed-initiative – many interactions will be simpler, not requiring any negotiation.

Why Use Mixed-Initiative and Ontology?

Since the beginning of computer gaming, designers have been struggling with the problem of interactivity: how do we give the player rich interactivity with the game world? Here we use the phrase, 'rich interactivity' to mean that the world responds in complex, semi-predictable ways to a suitably wide range of possible player actions. Most successful games display rich interactivity within certain limited domains. For example, in real-time strategy games the player can undertake a wide range of strategies and tactics, which the opponent(s) will respond to in ways that are only somewhat predictable. The richness of interaction in a game is limited by the time the player has to plan and execute complex interactions – in a first-person shooter, for example, tactics are chosen in fractions of a second, so gaining skill is a matter of tuning reflexes rather than developing complex strategies. These are solutions to the problem of interactivity: the challenge facing the game industry is providing new types of game play, and in general, the least explored areas of game play are in those areas where complex interactivity less limited by time constraints is made possible. Enabling game play that is centered on social interaction and player-driven narrative (one such less explored area) will be the main focus of this paper. Games that focus on meaningful social interactions represent a largely unexplored area for commercial game development, and may allow the market for computer games created by the game industry to broaden beyond the hardcore gamer.

Early text-based games often forced the user to guess which actions were possible, or at least guess the correct command to perform an action. Later games used an exclusively graphical representation of the game world, which made it easier to tell what was happening, but much harder to create a large set of possible actions for the player. In most games, especially in the first-person shooter and role-playing genres, player's interactions with the game world and characters are limited to either following a scripted dialog, or attacking the character, and players usually prefer the latter, probably because it provides more interactivity. The game world is largely static, responding only to the user's actions. Open-ended game play has been created for many games, but with the exceptions of simulations (flight simulators, *SimCity*, etc.,) there are serious limitations: player's social interactions with agents in the world remain scripted, and usually, the most interesting elements of role-playing games are still these scripted plots, simply because the open-ended activities the player can undertake are boring and predictable.

Another serious problem game developers are facing is the ever-increasing costs of game development. These costs stem from the expectations that computer game players have for better and better game content: graphics, stories, and game play are all expected to improve with each new release. These rising costs inhibit innovation in the industry. (Costikyan 2005) Mixed-initiative (MI) theory, in combination with ontology, can point the way to a method of game design that separates the actions of individual game agents from the game engine's code, allowing for the development of AI solutions somewhat independent from the specific game world. This is not a new concept: Magerko developed the Haunt 2 game based on an ontology of interactions. (Magerko and Laird 2003) Ontology permits us to encode a syntax for MI interaction between game agents that is largely independent of implementation, permitting reusability and reducing the complexity of the system by decoupling the logic of interaction from its implementation, much as some games (e.g. Half-Life) have decoupled their physics engine from their game engine. This is different from the methods employed in projects such as the Soar QuakeBot project, where an AI was implemented on top of existing game play (Laird and Duchi, 2000) – to support an MI/ontology approach, game engines will need to be designed specifically with this approach in mind.



With regards to the visual representation of the game world, the graphics engine can be tasked with representing the game state based on the ontological knowledge representation. In this way the ontological representation of the game world becomes the core model of the game world, with the behavior of AI and graphical representation both varying flexibly when that model changes.

Up to this point, mixed-initiative theory has been applied primarily to educational games, but a mixed-initiative approach can benefit the art and science of computer games in general. This paper examines how the concept of mixed-initiative interactions implemented through ontology can be applied to game programming to enable a more dynamic world with a wide range of interactive responses to the player's actions. It also explores the significant challenges in designing game play for a mixed-initiative game.

Mixed Initiative in Games: The State of the Art

Mixed-initiative interactions have been implicitly implemented in mass-market games in bargaining systems, helper agents, and 'simulation' games where players use mixed-initiative interactions to generate content - although mixed-initiative theory has not been applied to these genres.

Bargaining systems allow the user to negotiate with a computer agent for some trade: different types of resources might be exchanged, or services might be bought. In true mixed-initiative bargaining systems, each computer agent has its own goals, and may choose not to bargain for items the user wants to trade. A game that implements a bargaining system like this is *Freelancer*.

'Helper agents' also use mixed-initiative. An example is the game *Black & White*[1] that provides the player with a 'creature' (a giant ape, lion, or other animal) that can learn to imitate the player. (Crossan and Lessard 2005) The player can give the creature general commands, such as 'provide resources to this town', and correct the creature if it misbehaves by punishing it. The creature can try to attract your attention by following your cursor around the map, or even by intentionally misbehaving - an example of how an agent can attempt to initiate an interaction. Fail to correctly train your creature, or act unpredictably so that it cannot learn from you, and it may choose to disregard you entirely, often leaving a trail of destruction wherever it goes. In other words, the creature has its own goals, and if you aren't fulfilling them, it may choose to take independent action.

[1] http://www.lionhead.com/bw/

The most successful use of mixed-initiative in games to date has been in games designed by Will Wright. Wright has pioneered the area of player-driven content creation with *The Sims,* a 'simulation' of real life that allowed users to create and define their own characters through their interactions with the game world – developing their characters through their economic and social decisions. (Thompson 2003) By giving characters some autonomy, *The Sims* provides the player with a choice as to how much control they want over the characters – some players might be more interested in micro-managing their characters than others, or might prefer to focus on one character: whatever their preferences, the characters did their best to make their own decisions when not controlled by the player. In other words, the use of mixed-initiative gave players much more choice in how they played the game, and likely was a key factor in its success.

Applying Mixed-Initiative to Computer Games

Mixed-initiative approaches can be applied to several currently challenging areas of computer game design, including several types of agent AI, and automation. They can also help make possible more novel forms of game play.

One area in which the state of AI in games has been steadily (if slowly) progressing is automation of micromanagement in real-time strategy (RTS) games. Game playing in RTS games can be divided into two stages, micromanagement (building up resources and abilities and interacting with individual units) and macro-management (actions such as an attack that are directed against the enemy and are critical to the outcome of the game). Micromanagement is often considered tedious and steps taken to automate those functions help gamers focus more attention to the macro-management stages where critical interactions take place. Up to this point, automation efforts have not been mixed-initiative: AI techniques have been advanced at the level of the game's units (or 'agents'), and these agents often leave a lot to be desired since they are unable to perceive higher-level strategies. As we look at the evolution of games, we find that the amount of micromanagement required is gradually reducing in each generation of games. For example, in Age of Empires – the first of a series of RTS games - farms could not be reseeded and there were no queues for unit production leading to heavy intervention in the economy building stage and lesser in the actual fighting stage. This was remedied in *Age of Kings* (*Age of Empires* 2) and *Age of Conquerors* (its expansion pack), where many tedious functions were automated. *Age of Mythology*, its successor, automated almost all resource collection functions, and it is likely that as games develop, the entire focus of the game will move to further emphasize the macro-management process. For this to be possible, 'helper' agents capable of understanding more advanced tactics and strategies will be necessary, and it is here that mixed-initiative (MI) holds a lot of promise. MI is needed since gamers will not want to fully sacrifice their control over their micromanagement strategy, and at the same time want more automation of this area. An MI agent can provide this flexibility, taking action when appropriate, and asking the user for input or letting the user take control when necessary. This MI automation agent would greatly decrease the cognitive load by handling micro functions and at the same time allow for adaptation to a user's strategies, helping the user to concentrate on their macro-management strategies.

Finally, the idea of creating a dynamic and responsive game world where players can create their own goals and define their own characters has been around for a long time, although it has not been implemented in many commercially successful games up to this point. Some progress in this area has come from interactive fiction, although games in this genre have not been commercially successful. Early games such as *Zork* presented somewhat interactive stories through a text-based interface, but the rise of high-quality graphics and complexities in implementing an interactive story in a graphical environment ended commercial exploration of this field. (Murray, 1997) Chris Crawford, a longtime proponent of interactive fiction, has suggested that a "little language" - a language that contains only those concepts and relationships that the game world contains – can provide players with interactive fiction: this idea seems similar to the concept of domain-specific ontology. (Crawford 1993) However, mixed-initiative ideas have not generally been implemented in interactive fiction either, which generally remains tied to the idea of interactive dialog as a back-and-forth process: Crawford describes a conversation as a looping of the "listen, think, and speak" cycle. (Crawford 2005)

All of these problems represent areas where MI can be helpful. This paper will focus on a possible architecture for creating a flexible and powerful system for agent interactions in a game world, through the use of a mixed-initiative system based on a game-specific ontology representing game-world entities and their interactions.

A Mixed-Initiative Framework for Interaction through Ontology

The first step in defining a framework for mixed-initiative interactions is to define a means for knowledge representation. There are various knowledge representation techniques that have been explored in the field of computer games. DeSmedt et. al (1999) present an ontological approach to knowledge representation for NPCs, which the authors call conversational agents. Other researchers, such as Pisan (2000), present classification as a character building and knowledge acquisition tool. Narayek. et.al (2002) discuss intelligent agent planning in computer games for character representation. Magerko developed the Haunt 2 game based on an ontology of interactions. (Magerko and Laird 2003)

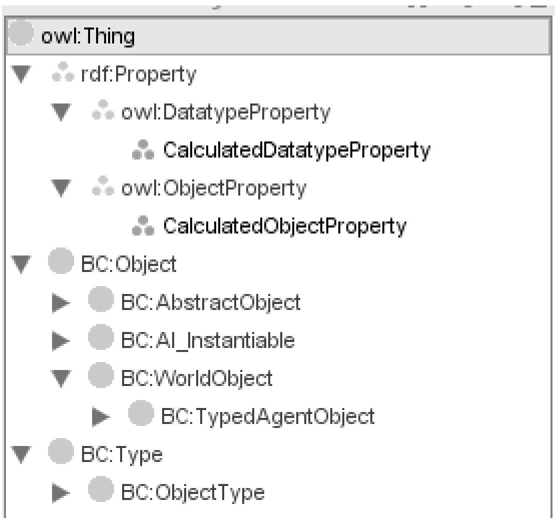
We propose that to support mixed-initiative interactions, a domain-specific ontology be used for knowledge representation, with a focus on game world objects as agents with their own interpretation of the world. Ontology allows us to define both the properties of various objects in the game world, and the relationships between them. This means that we can define all the game world entities, and all their properties and relationships to other entities. Sets of interactions common to given entity types can be defined in the ontology. The purpose of modeling the game world in ontology is three-fold: firstly, the ontology can be defined in a format (e.g. the Ontology Web Language) that is separate from the game. This means that game world rules and data can be reused in later versions or even somewhat different games. Secondly, and more importantly, ontology allows us to quickly define and modify the interactions available to agents in the game world, making the design of complex behavior such as mixed-initiative interactions more tractable. In other words, by providing a higher level of abstraction for agent interactions (and encoding world knowledge at that level as well) we separate what an agent can do (game world logic) from its decision-making processes (AI). Finally, using ontology makes building goal seeking agent AI easier – and goal seeking is an essential part of developing realistic mixed-initiative agent behavior.

We will now consider an example of how an ontology supporting mixed-initiative could be built to capture game world logic. First, an ontology modeling the game world's entities is developed: for example, an RPG's ontology would include all the races, weapons, armor, etc., and their attributes. Each actual in-game object that could be interacted with (i.e. probably excluding terrain, sky, etc.) would be modeled. Second, these general classes of objects are given interactions available to them: objects can be picked up, weapons can be attacked with, etc. Each interaction will have various properties specifying what is necessary to complete the interaction: for example, to execute an 'attack with a weapon' interaction, there will need to be an agent attacking, an object being attacked, and a weapon being attacked with. This is of course similar to an object-oriented approach – the key difference is that our ontology is available for use by the agent AI.

So far, we have not considered mixed-initiative. This is where the usefulness of our ontology becomes apparent. By modeling our interactions through ontology, we have provided the agents with a great deal of information on the objects and interactions they have available to them. Mixed-initiative agents must have goals, and in order to use goal-based reasoning, they need well-structured data about the world around them - which ontology provides.

We have considered the basic classes of knowledge the agent will have access to in the ontology. Actual in-game objects will be instantiations of those classes. These instantiations will then be 'sensed' by agents. This information will be interpreted by the agents, and filtered based on the senses they have available to them, their current knowledge, and the time since they directly sensed it (for example, an agent that didn't have knowledge about weapons might be unable to remember what weapon another agent was carrying later on, or if it was long enough ago, might have forgotten.) In this way, agents create their own unique perspective on the world. This is Isla and Blumberg's concept of *sensory honesty*: not only should an agent perceive the world according to the senses they are represented to the player has having, it should also have its own internal model of the world, which may differ from reality and/or that of another agent. (Isla and Blumberg, 2002) Sensory honesty becomes even more vitally important when attempting to build interesting behavior through MII, since the agent's biases and perceptions will persist as part of the game world's story, and any consistent flaw in the agent's sensory model will be very likely to become part of this story. For example, if an agent's AI is programmed to 'cheat', and perceive events further away then they would be expected to sense, the agent may then inform a player of the event, even though the agent should logically be unaware of it. In an extreme case, this could extend to agents relating the actions of agents they could never logically have met in the game world. In other words, when designing a system where knowledge-driven behavior is a goal, it is important to ensure that the perception system it will emerge from is consistent with the player's assumptions about that system. Finally, since agents have an ontology-based representation of the world, they can transmit it to other agents with the same ontology, and allow that information to gracefully degrade when other agents do not support parts of the ontology.

Having defined the design parameters for a mixed-initiative ontology-based game engine, we can make some initial suggestions as to what such a game engine might look like. We will use as an example the design of a game engine currently under development by some of the paper's authors. First, we can define a base ontology specifying the information needed by the base game engine and graphics engine, such as the positions of objects, their heading, etc. A game-specific ontology is layered on top of this, providing agent attributes and interactions. Properties are classified into two types: basic and calculated, with calculated properties being those derived from basic properties, and read-only. The dependencies for calculated properties are specified. This allows for the updating of dependent properties when the properties they are dependent on change. Calculated properties allow us to embed low-level logic such as path-finding into the ontology, and build properties that require calculations directly into the ontological model.



Taxonomic view of base game ontology in Protégé

Once the ontology has been defined, it is built into classes in the game language (in this case, C++.) These classes provide quick, efficient access to the game objects. The classes allow for two types of instantiation: real, and virtual. Real instantiations represent actual objects in the game world, whereas virtual instantiations represent agent knowledge about the world. In order to enable efficient AI solutions, state listeners can be plugged into any set of real instantiations to watch for significant state transitions. This means that an AI can watch for changes to the world state that interest it, without needing to scan large amounts of ontological data every cycle.

In summary, we propose the use of an ontological representation of knowledge, with the additional feature that it support the *transfer* of knowledge between agents through MI interactions, and be designed to allow agents to build rules and relationships about their world as a group. Through ontological representation, we are able to capture the contextual meaning of the content required for dynamic creation of agents' dialogues and other interactions. MI interactions will be built on the same ontology, eliminating the need for translations between interactions and those interactions' representation in the agents' knowledge representation.

Mixed-Initiative and Agent AI

A framework for mixed-initiative interactions in an open-ended role-playing game world will need to support not only one-to-one interactions, but also one-to-many interactions (e.g. shouting), and many to many interactions whose handling will be prioritized by the agent receiving them. Fortunately, the challenges to MII-interactions between teams of agents are actually an advantage here, since if implemented correctly, agents' failures to communicate and cooperate will provide as character definition and narrative content as their successes. For example, the challenges inherent in adjustable autonomy such as the *team decision challenge* (Scerri, Pynadath and Tambe, 2001), as well as other challenges related to the sharing of initiative can become part of the definition of an agent's behavior, since optimal helpfulness is not necessary in most situations. An example would be that a character with a low tendency to give up initiative and take commands from other characters would seem more independent. Variables determining an agent's MI strategies can become part of its character.

The flexibility of supporting multi-agent interactions and the handling of many such interactions simultaneously is necessary since the game world will require these types of interactions. Ideally, the same framework will handle interactions between all game objects, even those as simple as an agent picking up an object. For the purposes of this system, any game world object could be considered a potential participant (or 'agent') in the mixed-initiative interaction system. This will allow for the modification of behaviors of game objects to be encapsulated through the interactions interface – for example, a chair could be given a 'broken' attribute, so that when another agent tried to 'sit' in it, the interaction was refused – this without modifying the agent's AI at all. Assuming the agent's AI is smart enough to be able to handle the possibility that sitting in a chair will not work (i.e. it may be occupied), the new property and behavior will not require modification to the agent. A truly advanced agent might even be able to ask the chair what property or properties prevented it from sitting down, receive back the portion of the instantiated ontology specifying that the chair was broken, and determine from a rule-set that it needed to repair the chair – an entire new behavior implemented without changing the code base or the agent's AI routines.

The most general case of the MI interaction system would be when it is in the process of perceiving input from surrounding agents (players, NPCs, etc.) Some input may trigger the creation of a context (such as a 'conversation' between a set of agents.) The agent's system can then place those objects into a new context. A context could include some rules or guidelines for how interaction in that context could take place, similar to the types of interactions defined by Guinn: for example, SingleSelection (where one agent takes control of the context's interactions) or Continuous (where agents' right to control the context is reassessed continuously during the interaction) modes could be indicated (either explicitly or through variable changes in behavior, such as weightings inhibiting certain interactions, much as social mores inhibit possible real-life actions.) (Guinn, 1996) At the same time, the system should probably be inherently agent-oriented, where interactions will be prompted by the agent's goals – an agent without goals (or currently without the means to further them) will not initiate interactions. We emphasise that the 'interactions' possible are not limited to speech – body language and actions will also be considered.

The degree to which a MI approach to agent interactions in an open-ended world can be successful rests in part on technological limitations: how many agents with what level of cognitive ability can a single PC support using the fraction of its resources that a game can allocate to AI? Actual implementations need to be tested in order to determine what types of MI games can be implemented given these constraints.

Designing Ontology-based Agent AI

Completely addressing the design of an ontology-based AI supporting mixed-initiative in games is beyond the scope of this paper, however, some design patterns can be suggested. First, the AI system will usually need two components: a planning component that can find ways to transform the current world state into the goal states of the agent through the interactions defined in the ontology, and a component that can choose the best entities to instantiate those interactions. For example, if an agent wants to kill an enemy, it will need to determine the steps necessary to do this, such as:

1. Determine where the enemy is (talk to other agents or search,)
2. Get a weapon (find a store, choose the best weapon given the amount of money the agent has, its other equipment, and its abilities.)
3. Go to where the enemy is.
4. Attack the enemy with the weapon.

And so on until the objective is achieved. Two types of AI are needed here: qualitative goal-oriented planning in order to determine which actions to take, and quantitative judgment in order to determine which people to talk to, which weapon to buy, and so on. While these sorts of decisions could be hard-coded, both qualitative planning and quantitative judgment can benefit from a set of rules that capture causal relationships and algorithms for quantitative judgments in the most general terms that is possible. For example, a rule might be "If you know the position of an object you are looking for to within 50 meters, search for it, otherwise ask other people where it is." Developing a set of rules that creates intelligent agent behavior in every situation is difficult, so agents should probably be constrained from attempting anything too mentally challenging: for example, any goal which requires them to travel outside of their 'home' area might be culled, or an upper limit on the number of steps in a plan could be enforced. It will usually be preferable that agents do nothing rather than doing something stupid: after all, players are used to computer agents that rarely take initiative beyond simple actions. It should be noted that agents do not have to be given these planning abilities – an agent could be designed to take action only when all conditions are set for a particular action to bring about a particular goal – but without them, their motivations will be less obvious and seem less convincing to the player.

Mixed-Initiative and Game Play

There are a number of challenges to creating compelling game play with mixed-initiative. This section considers a few of them.

The goal of developing a mixed-initiative framework is to enable believable interactions between agents (both computer and human.) However, this means that the amount of varied and interesting behavior in computer agents will be dependent on the complexity of their AI, and the complexity of the game world. It is easy for agents to fall into repetitive, annoying, or simply stupid goal-oriented behavior, bothering the human agents. Careful design will be required to minimize this risk.

Another challenge is integrating a story line with mixed-initiative. Agents with their own goals and priorities must still provide the right information to the player(s) at the right time. At the same time, the player(s) should not have to mine conversations for useful information – a player should be able to simply ask a question. This raises the problem of translation between the vocabulary of the ontology and the desired human language – how can the many different fragments of knowledge an agent may want to communicate be translated into the correct language and idiom for the game world? Initial experiments in MII should probably limit interactions to atomic ones that can be translated using simple rules, without needing to take context into consideration. In other words, true conversation is an unrealistic goal: for example, an agent should not have to understand a pronoun based on previous interactions with the player.

There are undoubtedly further challenges to designing mixed-initiative game play that will be encountered. Experimentation and subsequent analysis will probably be the most useful method of confirming them.

Future Work

This paper has laid out some basic suggestions for implementation of a mixed-initiative ontology-based agent interaction system for computer games. Further analysis of this problem will be valuable. Even more so, concrete experimentation in implementing ontology-based interaction systems could provide data on the feasibility and complexity of the problem.

Conclusion

We have established that MII and ontology can and should be considered in the improvement of interactivity in computer games. We have analyzed the requirements and possible architecture of an ontology-based mixed-initiative system for agent interaction, and developed some guidelines for implementing such a system. In general, these guidelines recommend that game play and game world logic be defined through ontology, with agent AI encapsulated for each agent so that agents' AI can be developed and enhanced independently. This model also allows agents to develop their own models of the world, providing a realistic subjectivity to their views, and enriching the variety of interactions the player(s) experience.

Acknowledgments

This research was funded by the LearningKit project (SSHRC-INE) and the LORNET project (NSERC).

References

Chandrasekaran, B., Josephson, J. and Benjamin, V. 1999. What are Ontologies and Why Do We Need Them? *IEEE Intelligent Systems*, 14, 1, 20-26.

Costikyan, G. 2005. *Burning Down the House: Game Developers Rant*. Game Developers Conference 2005, San Francisco, USA.

Crawford, C. 2005. *Chris Crawford on Interactive Storytelling*. Berkley, CA: New Riders.

Crawford, C. 1993. A Better Metaphor for Game Design: Conversation. Journal of Computer Game Design. 6 Retrieved from http://www.erasmatazz.com/library/JCGD_Volume_6/Conversational_Metaphor.html

DeSmedt, B. & Loritz, D. 1999. Can We Talk? in *Papers from the AAAI 1999 Spring Symposium on Artificial Intelligence and Computer Games*, Technical Report SS-99-02, AAAI Press, pp 28-31

Guinn, C. 1996. Mechanisms for Mixed-Initiative Human-Computer Collaborative Discourse. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics.* 278 – 285. Morristown, NJ, USA: Association for Computational Linguistics

Hearst, M.A. 1999. *Trends and Controversies - Mixed-initiative interaction.* IEEE Intelligent Systems, September/October 1999, 14 - 24.

Isla, D. & Blumberg, B. 2002. New Challenges for Character-Based AI for Games *in Papers from the AAAI 2002 Spring Symposium on Artificial Intelligence and Interactive Entertainment, Technical Report SS-02-01*, AAAI Press, pp 41-45.

Laird, J.E., & Duchi, J.C. 2000. *Creating Human-like Synthetic Characters with Multiple Skill Levels: A Case Study using the Soar Quakebot.* AAAI 2000 Fall Symposium Series: Simulating Human Agents, November 2000. Retrieved September 10, 2005 from the World Wide Web: http://ai.eecs.umich.edu/people/laird/papers/human-level-quake.pdf

Murray, J. 1997. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace.* New York, NY: The Free Press.

Nareyek, A. 2002. Intelligent Agents for Computer Games, in Marsland, T. A., and Frank, I. (eds.), *Computers and Games, Second International Conference, CG 2000*, Springer LNCS 2063, 414-422.

Pisan,Y. 2000. Building Characters: A Form of Knowledge Acquisition *in Papers from the AAAI 2000 Spring Symposium on Artificial Intelligence and Interactive Entertainment, Technical Report SS-00-02*, AAAI Press, pp 66-68.

Scerri, P., Pynadath, D., & Tambe, M. 2001. Adjustable Autonomy in Real-World Multi-Agent Environments. In *Proceedings of the fifth international conference on Autonomous agents*. 300-307. New York, NY, USA: ACM Press.

Thompson, C. 2003. Suburban rhapsody. *Psychology Today*. New York: Nov/Dec 2003. 36(6):32